

"Express Mail" Label No.: EL 848970505 US

Date of Deposit: October 20, 2003

Attorney Docket No.14896US02

SYSTEM WITH REQUIRED ENHANCEMENTS TO
SYNCML DM ENVIRONMENT TO SUPPORT FIRMWARE UPDATES

RELATED APPLICATIONS

[0001] This patent application makes reference to, claims priority to and claims benefit from United States Provisional Patent Application Serial No. 60/419,903, entitled "System with Required Enhancements to SyncML DM Environment to Support Firmware Updates," filed on October 21, 2002.

[0002] The complete subject matter of the above-referenced United States Patent Application is hereby incorporated herein by reference, in its entirety. In addition, this application makes reference to United States Provisional Patent Application Serial No. 60/249,606, entitled "System and Method for Updating and Distributing Information", filed November 17, 2000, and International Patent Application Publication No. WO 02/41147 A1, entitled "Systems And Methods For Updating And Distributing Information," publication date March 23, 2002, the complete subject matter of each of which is hereby incorporated herein by reference, in its entirety.

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0003] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[0004] [Not Applicable]

FIELD OF THE INVENTION

[0005] The present invention relates generally to the generation and dissemination of update packages for software and/or firmware employing the Synchronization Markup Language (SyncML) device management (DM) specifications and, more specifically, to the enhancements to SyncML DM in terms of new commands that facilitate firmware updates from one version to another.

BACKGROUND OF THE INVENTION

[0006] Electronic devices, such as mobile phones and personal digital assistants (PDAs), often contain firmware and application software either provided by the manufacturer of the electronic devices, by telecommunication carriers, or by third parties. The firmware and application software often contain software bugs. New versions of the firmware and software are periodically released to fix the bugs or to introduce new features, or both.

[0007] SyncML, founded in February 2000, is an open industry standard for a common data synchronization protocol that facilitates the transport of network data and personal information across multiple networks, platforms and devices. Ericsson, IBM, Lotus, Matsushita, Motorola, Nokia, Starfish Software, Symbian, and Openwave sponsor the SyncML standard.

[0008] The SyncML standard was originally created to enable end users to seamlessly synchronize any type or format of network data with any mobile device. Since the creation of the data synchronization specification, the standards body has also developed the DM specification, which leverages the original specification. The two specifications are designed to be independent of the mobile device, network, and applications. For this purpose SyncML utilizes Extensible Markup Language (XML) technology.

[0009] The first specification to evolve from the SyncML initiative was centered on synchronization of networked information to a mobile device. The goal of the common synchronization protocol was to solve the problem of multiple proprietary data synchronization protocols that limited the use and availability of data. The data synchronization specification enables any networked data to work with any mobile device. For example, an end user could access personal information, such as an address book, from a handheld or PDA and the data would be accessible and current.

[0010] Synchronization has different semantics when applied to firmware updates. For example, generic synchronization attempts to make an instance of data that looks like X into an instance of data that looks like X' (X-prime), and vice versa, without regard to the availability of various other versions of the same data, such as X'' (X-prime-prime), etc.

Firmware updates, in general, require the specification of what the target version should be.

[0011] In addition, firmware updates differ from synchronization in that the carrier-side or server-side does not have an entity that is a firmware – it only contains data that may be generated/translated into a firmware.

[0012] The second specification from the SyncML initiative revolves around the remote management of mobile devices, or device management (DM). Device management is the generic term used for technology that allows third parties to carry out the procedures of configuring mobile devices on behalf of the end user. SyncML defines device management as a technology that enables customization, personalization, and servicing of personal devices.

[0013] The DM specification proposes that future releases will facilitate software distribution, parameter configuration, and device capability verification. According to the SyncML White Paper, the planned device management scope includes device configuration (e.g. modifying or reading operating parameters), software maintenance (e.g. reading from a device its current operating parameters, reading a list installed or running software, hardware configurations), and diagnostics (e.g. listening for alerts sent from a device, invoking local diagnostics on a device).

[0014] The initial specification for the device management standard was released in February 2002. That release describes, from a high level, the protocol, component description, management frameworks, and security involved with the device management process. The DM specification has alluded to the fact that one of the future features would be the capability to download software. Details of the "software download" process were not included in the latest specification, but the download specification is projected to be completed in 2003. The standard does not specify formats of or methods to create update packages (encoded information of modifications to firmware/software necessary for upgrading to a new or modified version of the firmware/software) or how to apply the updates within the device. It mainly deals with management of the meta-data and the process around it. As with the aforementioned data synchronization protocol, the DM protocol will be open, and network and device agnostic.

[0015] SyncML has been promoted by the wireless industry as a solution for data transfers between mobile handsets and service providers. Although flexible, the current SyncML DM specifications do not yet provide support for firmware updates.

[0016] There is a fundamental problem in supporting firmware updates employing SyncML DM-based software. There is no support in SyncML DM for firmware updates, as there are no commands or request structure that support setting of firmware version change indicators, power cycling, authentication/verification of software packages or firmware updates received, etc. In addition, the ability to request a software/firmware update in terms of source and target versions is essential but not currently supported in SyncML DM. Specifically, the semantics of retrieving a firmware update from a server on a carrier network is different from that of acquiring/retrieving generic data or content from a web site, and SyncML DM does not currently support these special semantics.

[0017] There is a fundamental asymmetry introduced into the client and server-sides of any SyncML-based solutions. Any “power cycling” command (needed to reinitialize a handset) that may be introduced into SyncML may apply only to the client-side, with SyncML core (or engines) on the server-sides configured to ignore such commands. In addition, any requests to update firmware may be restricted to the client-side.

[0018] Recent versions of SyncML DM do not have commands exclusively aimed at one side or the other, i.e. a command that is restricted to the client-side alone or the server-side alone. The SyncML DM management client may be a combination of a Download Agent with a security service and a resource manager that manages a tree of managed objects.

[0019] SyncML DM does not yet provide Update Agent (an agent for applying firmware updates in an electronic device) capabilities; it provides support for executing commands on the client-side. Such commands may include commands to Add, Copy, Delete, Exec, Get, Sequence, etc. that may be used to manipulate managed objects. Therefore, management objects may be added, copied, deleted, replaced, etc.

[0020] The SyncML management client may process XML, execute commands and transfer data. A SyncML DM client may comprise an XML parser that may employ SyncML DM elements and commands; an “engine” that may execute commands based on data retrieved from SyncML (XML) content; a tree or registry of management objects

that may be targets of commands executed by the engine; and a data transport layer that may be based on one of Hyper Text Transfer Protocol (HTTP), Object Exchange (OBEX), Wireless Session Protocol (WSP), etc.

[0021] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of ordinary skill in the art through comparison of such systems with the present invention.

BRIEF SUMMARY OF THE INVENTION

[0022] Aspects of the present invention may be seen in a system employing Synchronization Markup Language (SyncML) device management specifications to facilitate firmware updates in an electronic device. The system comprises at least one electronic device having a memory, at least a portion of the memory comprising non-volatile memory containing firmware; a SyncML server communicatively coupled to the electronic device, the server comprising an enhanced SyncML DM server software; and a SyncML DM client resident in the electronic device, wherein the SyncML DM client is capable of interpreting enhancements to the SyncML DM specifications for updating the firmware.

[0023] The electronic device comprises communication software that supports at least one data transport protocol; a security module; and at least one software function that provides access to proprietary parameters in the electronic device.

[0024] The SyncML DM client comprises message processing software that facilitates processing and executing of SyncML messages, commands, alerts, and notifications; a SyncML manager; a download software that facilitates the downloading of at least one firmware update package from the SyncML server; and an update software that facilitates the updating of firmware using the at least one firmware update package.

[0025] The method for updating firmware in the electronic device in the system employing enhancements to SyncML DM specifications comprises sending a SyncML based notification to the electronic device; facilitating communication between the electronic device and the SyncML server; receiving the notification by a SyncML DM client resident in the electronic device; parsing the notification; and sending the notification for user review and subsequent user input. The method further comprises initiating a firmware update based on an input by the user; sending the firmware update to a download agent in the electronic device; communicating an appropriate SyncML message to initiate download of an update package from the SyncML server; facilitating and analyzing a response from the SyncML server; verifying validity and authentication of the update package, if an update package is received as part of the response; and dispatching commands in the response to appropriate modules.

[0026] These and other features and advantages of the present invention may be appreciated from a review of the following detailed description of the present invention, along with the accompanying figures in which like reference numerals refer to like parts throughout.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0027] Fig. 1 illustrates a block diagram of an exemplary system with SyncML DM environment, in accordance with an embodiment of the present invention.

[0028] Fig. 2 illustrates an exemplary end-to-end architecture, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0029] Fig. 1 illustrates a block diagram of an exemplary system 105 with SyncML DM environment, in accordance with an embodiment of the present invention. The system 105 comprises a mobile handset 107 and a SyncML server 109. The mobile handset 107 and the SyncML server 109 may be connected via a communications link 145. The system 105 may employ enhancements to SyncML DM specifications to facilitate firmware updates on the mobile handset 107. The mobile handset 107 may comprise a SyncML management client 111, and a communication layer 113 that may provide one or more data transport protocols. In addition, the mobile handset 107 may comprise a security module 133, and device wrappers 135 that may provide access to configuration parameters, hardware elements, memory elements and User Interface (UI) modules. In an embodiment of the present invention, the mobile handset 107 may also comprise a SIM card reader 131 and a SIM card 129.

[0030] The SyncML server 109 may comprise a SyncML engine 137. In an embodiment of the present invention, the SyncML server 109 may also comprise a provisioning interface 139, which may provide an interface to one or more external service provisioning systems, a content database 141, which may provide access to content such as firmware update packages, and a notification manager 143, which may provide notification support to facilitate notification of mobile handset 107. The SyncML engine 137 may facilitate the creation and communication of SyncML based messages, notifications, update packages, etc. to the mobile handset 107. The SyncML engine 137 may also support parsing and execution of SyncML requests including the enhancements to SyncML DM introduced by the present invention.

[0031] The SyncML management client 111 may comprise a SyncML engine 115, a SyncML manager 123, a download agent 125 and an update agent 127.

[0032] The SyncML engine 115 may facilitate the processing and execution of SyncML messages, commands, alerts, notifications, etc. The SyncML engine 115 may comprise a parser 117 that may be used to parse incoming SyncML messages to retrieve commands and data, a command builder 119 that may be used to assemble SyncML messages, requests, notifications, alerts, etc., and a dispatcher 121 that may dispatch commands

received in SyncML based messages, alerts, etc. for execution by the SyncML management client 111 or other modules or applications in the mobile handset 107.

[0033] The SyncML manager 123 may provide support for managing the life-cycle of management objects, managing SyncML command execution, facilitating user interactions, etc. The download agent 125 may facilitate the download of update packages for software/firmware from the SyncML server 109 employing one or more data transport protocols supported by the communication layer 113. The update agent 127 may facilitate the update of the firmware/software of the mobile handset by employing one or more downloaded update packages.

[0034] In an embodiment of the present invention, the SyncML server 109 may send a SyncML-based notification, to the mobile handset 107, employing an external notification manager 143. The notification may indicate the availability of a firmware update package. The communication layer 113 may facilitate the communication between the mobile handset 107 and the SyncML server 109. The SyncML management client 111 may receive the notification message from the SyncML server 109. The SyncML management client 111 may employ the parser 117 in the SyncML engine 115 to parse the received SyncML message. The dispatcher 121 may dispatch the notification for user review and subsequent user input, employing device wrappers 135. The device wrappers 135 may be functions utilized to indirectly access proprietary information or code stored in hardware device (e.g., registers).

[0035] The dispatcher 121 may dispatch subsequent user-initiated update of firmware to the download agent 125. The download agent 125 may communicate an appropriate SyncML message assembled by the command builder 119, to initiate download of an update package from the SyncML server 109. The download agent 125 may also facilitate the download of a response from the SyncML server 109, the response subsequently being parsed by the parser 117 and analyzed by the SyncML manager 123 and the SyncML engine 115. If an update package is received from the SyncML server 109 as part of the response, the SyncML manager 123 may ensure validity and authentication of the update package, by employing the security module 133 and the device wrappers 135. The dispatcher 121 may dispatch the commands in the SyncML-based response to appropriate modules via device wrappers 135 or via the SyncML

manager 123. The commands may comprise commands such as, for example, a command for the verification of a received update package, a command for saving the update package in an appropriate management object, a command for initiating an update process by the update agent 127 and a command for subsequent notification of the result of update agent processing (success, failure, etc.).

[0036] In an embodiment of the present invention, the SyncML management client 111 may employ new commands for retrieving update packages for firmware updates and for updating the firmware based on appropriateness, security and authentication, employing fault tolerant update mechanism. The SyncML engine 137 in the SyncML server 109 may be capable of processing the new enhancement commands sent to it by the mobile handset 107.

[0037] The update packages may be accompanied by a header that contains, among other entries, a cyclic redundancy check (CRC) value employed in the verification of the authenticity of the received update packages. The verification of received update packages may involve computing CRC values and comparing them to reference CRC values provided in a header that accompanies the update packages. Other forms of verification and authentication based on specific entries in the header are also contemplated.

[0038] In an embodiment of the present invention, a command to communicate the change of SIM card 129 detected by the mobile handset 107 may be supported by the mobile handset 107. The SyncML engine 137 in the SyncML server 109 may be capable of processing such a command and acting upon it.

[0039] In an embodiment of the present invention, a client-side device such as, for example, a mobile handset 107, may incorporate a management client 111 that may comprise an update agent 127, a download agent 125 (simplified to ride on top of/employ one or more data transport protocols), a SyncML engine 115, and a SyncML manager 123. The SyncML engine 115 may comprise a SyncML dispatcher 121, a SyncML parser 117, and a SyncML command builder 119.

[0040] In an embodiment of the present invention, a communication layer 113 may be utilized for the download agent 125 to interact with a management server such as, for example, the SyncML server 109. The communication layer 113 may support such

function as, for example, protocolInitialization, openConnection(), closeConnection(), sendData(), and receiveData(), on which the download agent may depend.

[0041] In an embodiment of the present invention a security layer may be utilized. The security layer may be considered to be external to the management client.

[0042] Aspects of the present invention retain differentiation in the creation of the update package and the process of applying the update fault tolerantly, as allowed by the SyncML standard. An embodiment of the present invention may utilize a newly defined, extensible, XML-based Agent-to-Server meta-data protocol that adapts more easily to the SyncML DM "software download" standard, when it is ratified. A package query protocol associated with an embodiment of the present invention may follow the same extensibility and meta-data management principles seen in current SyncML standards. As a result, various embodiments of the present invention may easily comply with the SyncML standard.

[0043] SyncML DM provides a management client that is responsible for providing access to management objects. The management objects may be arranged in a management tree. The management client may be capable of manipulating the management objects and the management tree. In addition, a management tree may be comprised of management objects where each management object is a unit of code/data/application that may be accessed and/or manipulated.

[0044] The management client proposed by the SyncML standards is responsible for client-side support for SyncML based communications with management servers. Some of the duties associated with the management clients, include, for example, processing SyncML data, parsing, dispatching, and command building. A management client in an embodiment of the present invention may support dispatching and/or executing of SyncML DM commands such as, for example, alert, replace, status, add, copy, delete, exec, get, sequence, etc. The management client may also interact with a management server employing SyncML commands and structure using a communication toolkit. In addition, the management client proposed by the SyncML standards may support session initiation and termination.

[0045] The Alert command may be used to convey notifications, such as device management session requests, to the recipient. Such device management session requests

may be client initiated or server initiated. In addition, wireless applications protocol- (WAP) push-based notification may be a required feature in SyncML DM servers and clients, in an embodiment of the present invention.

[0046] SyncML DM clients may have to support security features such as hash message authentication code (HMAC), including basic authentication, challenge/responses, etc.

[0047] In an embodiment of the present invention, on the server-side, a device server may have its own SyncML engine to process user requests as well as to provide requested data to the mobile handsets in SyncML format. The device server may also be responsible for implementing the required security services, inserting security challenges in SyncML communications if necessary.

[0048] In an embodiment of the present invention, to be SyncML-compliant, an engine that executes SyncML DM commands may be provided. A tree of management objects may be created and maintained by the management client.

[0049] An embodiment of the present invention may comprise a SyncML management client. As a result, in addition to the download agent and the update agent, an embodiment of the present invention may employ XML parsers, encryption libraries, etc. in a mobile handset in order to assemble XML data/requests to be sent as SyncML DM commands/Alerts/Requests, parse any received XML data if data is returned as XML by the management server, and incorporate security mechanisms that may be used (Message Digest 5 (MD5), HMAC, etc.).

[0050] An embodiment of the present invention may enhance its set of device wrapper functions to be able to retrieve all the XML elements sent to the management server. The management client in such an embodiment may comprise its own download agent.

[0051] In accordance with the present invention, mobile handsets, which provide support for XML, may be multi-threaded. In a multi-threaded environment, it may not be appropriate to assume that an update process can immediately follow a download of an update package, because other applications may be running concurrently. As a result, an embodiment of the present invention may provide separate download and update processes.

[0052] An embodiment of the present invention may support reboot commands, reset commands, and power cycling by providing a location for storing downloaded update packages. An embodiment of the present invention may provide such a location by specifying a managed object for firmware updates.

[0053] An embodiment of the present invention may also provide an update package in a firmware updating service that comes up automatically during a power cycle/reboot. A SyncML DM command set enhancement that initiates a power cycling/reboot may be provided in an embodiment of the present invention. Another command enhancement may store a downloaded firmware update into an associated managed object. Yet another command enhancement may verify the authenticity of the firmware update package based on configurations and user profiles.

[0054] An embodiment of the present invention may utilize enhancement commands such as, for example, GetFirmwareUpdate, VerifyFirmwareUpdate, SaveFirmwareUpdate, ApplyFirmwareUpdate, ConfirmFirmwareUpdate, and SIMCardChange, explained below. An embodiment of the present invention may add these enhancement commands to the set of SyncML DM Protocol Command Elements as enhancements.

[0055] In an embodiment of the present invention, the SyncML DM protocol allows the enhancement commands to be executed on nodes of the management tree in the mobile device 107. The SyncML device management may be the means by which the management tree is manipulated by the SyncML DM server. In an embodiment of the present invention, each node is addressed by a unique uniform resource identifier (URI). In SyncML DM protocol, the target and source of a command are identified by the target and source elements.

[0056] In an embodiment of the present invention, the SyncML management server may employ the exec command to invoke the enhancement commands associated with the firmware updates in the mobile handset. The exec command may launch a process that initiates the firmware update download in accordance with the parameters provided in the exec command. The implementation of the exec command may ensure that the entire firmware update file download is completed prior to starting any code rewrites on the device.

[0057] The command GetFirmwareUpdate may create a new management object that refers to an update package of the firmware. This command may specify an item element to be retrieved; the item element may be also specified with an associated target element. The firmware update retrieved may be saved as a management object.

[0058] The GetFirmwareUpdate command, in an embodiment of the present invention, may verify whether an update package already exists in the mobile handset before initiating a new download, in order to avoid an unnecessary download.

[0059] In an embodiment of the present invention, the GetFirmwareUpdate command may provide the functionality of the SyncML DM GET command, in addition to providing the support to verify whether the retrieval is necessary before conducting it.

[0060] In an embodiment of the present invention, the SyncML DM server may employ the GetFirmwareUpdate command to download the firmware update package from a given location such as a uniform resource locator (URL). The GetFirmwareUpdate command may be executed by the SyncML DM protocol using the exemplary XML code shown below:

```
<GetFirmwareUpdate>
  <CmdID>3</CmdID>
  <Item>
    <Target>
      <LocURI> 'Managed Object where update package stored is specified here' </LocURI>
    </Target>
    <Source>
      <LocURI>'URL of update package is specified here' </LocURI>
    </Source>
    <Data>DL_URL=AnyURL</Data>
    <Data>DL_Prompt=[0|1]</Data> <!-- 0 is no prompting, 1 is with prompting -->
  </Item>
</GetFirmwareUpdate>
```

[0061] The Target element of the GetFirmwareUpdate command specifies where in the management tree the downloaded update package gets inserted. The Data element DL_URL specifies the source URL for the download of the firmware update package. In an embodiment of the present invention, the command may be executed with a user prompt, in which case an appropriate value is employed in the Data element for DL_Prompt value. Other Data and/or Target elements may be also used.

[0062] In an embodiment of the present invention, other forms of download of the firmware update package may be utilized. For example, the following Download command, which may be a generic form of the GetFirmwareUpdate command, may be used with firmware as well as software and other forms of generic downloads:

```

<Download>
  <CmdID>3</CmdID>
  <Item>
    <Target>
      <LocURI> 'URI of a node for the storage of update package is
        specified here, or a node that represents the target of the update' </LocURI>
    </Target>
    <Source>
      <LocURI> 'URL of update package is specified here' </LocURI>
    </Source>
    <Data>DL_URL=AnyURL</Data>
    <Data>DL_Prompt=[0|1]</Data> <!-- 0 is no prompting, 1 is with prompting -->
  </Item>
</Download>

```

[0063] An embodiment of the present invention may support a VerifyFirmwareUpdate command to verify the authenticity and appropriateness of a downloaded firmware update. Appropriate results/errors may be communicated to the SyncML management client when appropriate. The VerifyFirmwareUpdate command may be executed by the SyncML DM protocol using the exemplary XML code shown below:

```

<VerifyFirmwareUpdate>
  <CmdID>3</CmdID>
  <Item>
    <Target>
      <LocURI> ./FWUpdate/x/PkgStatus</LocURI>
    </Target>
    <Source>
      <LocURI> ./FWUpdate/x/PkgURL</LocURI>
    </Source>
  </Item>
</VerifyFirmwareUpdate>

```

[0064] The VerifyFirmwareUpdate command may verify the specified update package. The update package may be specified using the Source element, and the result of such verification may be populated into a Target URI. Other Target and/or Source elements may also be used.

[0065] An embodiment of the present invention may support a SaveFirmwareUpdate command to save the firmware update in an appropriate management object and/or FLASH segments in memory. The SaveFirmwareUpdate command, in an embodiment of the present invention, may provide support for the functionality of the SyncML DM

Replace or Add commands. The SaveFirmwareUpdate command may be used to ensure that an update agent sets an “update” bit in FLASH for a subsequent update process. The SaveFirmwareUpdate command may be executed by the SyncML DM protocol using the exemplary XML code shown below:

```
<SaveFirmwareUpdate>
  <CmdID>3</CmdID>
  <Item>
    <Source>
      <LocURI>./FWUpdate/x/PkgData</LocURI>
    </Source>
    <Data>Address=0x00089F</Data> <!--Address specified as hex value-->
  </Item>
</SaveFirmwareUpdate>
```

[0066] The SaveFirmwareUpdate command facilitates saving of the update package in specific address locations that may be implementation-dependent. In the above example, binary data from the management tree located in the mobile handset at the location ./FWUpdate/x/PkgData is saved at the address specified in the Data element for ‘Address’ parameter. Other Source and/or Data elements may also be used.

[0067] An embodiment of the present invention may support an ApplyFirmwareUpdate command to take a target firmware update management object and launch an update process. The update process may involve power cycling the mobile handset. The result of taking a target firmware update management object and launching an update process may be communicated only after the power cycling step has been executed. The ApplyFirmwareUpdate command may be executed by the SyncML DM protocol using the exemplary XML code shown below:

```
<ApplyFirmwareUpdate>
  <CmdID>3</CmdID>
  <Item>
    <Source>
      <LocURI>./FwUpdate/x/PkgURL</LocURI>
    </Source>
    <Data>argument1</Data>
    <Data>argument2</Data>
    <Data>argument3</Data>
  </Item>
</ApplyFirmwareUpdate>
```

[0068] The ApplyFirmwareUpdate command facilitates the invocation of an update agent. The ApplyFirmwareUpdate command specifies the Source element as a URI in the

management tree, and specifies optional prompts. Other Source and/or Data elements may also be used.

[0069] In an embodiment of the present invention, other Update commands may be utilized for updating firmware with a firmware update package. For example, the following Update command, which may be a generic form of the ApplyFirmwareUpdate command, may be used:

```
<Update>
  <CmdID>3</CmdID>
  <Item>
    <Target>
      <LocURI>./.../Application/ ApplicationXXX</LocURI>
    </Target>
    <Source>
      <LocURI>./FwUpdate/x/PkgURL</LocURI>
    </Source>
    <Data>argument1</Data>
    <Data>argument2</Data>
    <Data>argument3</Data>
  </Item>
</Update>
```

[0070] In the Update command, Target specifies the node in the management tree that represents the Application or component to be updated. The Source specifies the URL from which the update package(s) are to be downloaded or retrieved. A node that provides an update package that may have been previously downloaded, or may be part of the management tree, may also be employed.

[0071] An embodiment of the present invention may support a ConfirmFirmwareUpdate command to confirm the status of the previously executed firmware update event. The ability to confirm successful completion of a firmware update may be used to initiate billing activities on the carrier-side. The ConfirmFirmwareUpdate command, in an embodiment of the present invention, may support the functionality of a SyncML DM generic update confirmation command/notification, which may be utilized to indicate a successful completion of update events, specifically to initiate possible billing activities.

[0072] When a SIM card changes on the mobile handset side, the nature of the service may change on the carrier-side. An embodiment of the present invention may support a SIMCardChange command to facilitate the notification of service-level changes with possible changes to billing events.

[0073] **Fig. 2** illustrates an exemplary end-to-end architecture, in accordance with an embodiment of the present invention. The solution depicted in **Fig. 2** is compliant with SyncML specifications.

[0074] A mobile handset 219 may comprise firmware. The mobile handset 219 may be communicatively coupled with a SyncML DM server 209 via a communication link 213. The communication link between the SyncML DM server 209 and the mobile handset 219 may utilize a SyncML protocol 213. The SyncML DM server 209 may receive a firmware update package 207 from the generator 205 that generated the update package 207 utilizing an old (original) version of the firmware 201 and a new (target) version of the firmware 203.

[0075] In an embodiment of the present invention, the SyncML DM server 209 may be used to store the update packages 207, to manage delivery of the update packages 207, and to maintain system security. The enhancement commands described hereinabove enable an enhanced SyncML DM server 209 to provide the firmware updating service, which is not supported by the commands available using a server compliant only with the existing SyncML DM standards.

[0076] In an embodiment of the present invention, the mobile handset 219 may comprise a SyncML DM download agent 215, and an update agent 217. The SyncML DM download agent 215 may be responsible for initializing, transferring, and checking the update package 207 sent by the SyncML DM server 209. The update agent 217 may then apply the updates to the firmware in a fault tolerant manner in accordance with an embodiment of the present invention.

[0077] While the present invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the present invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the present invention without departing from its scope. Therefore, it is intended that the present invention not be limited to the particular embodiment disclosed, but that the present invention will include all embodiments falling within the scope of the appended claims.